Software Tracing of Virtual Hardware using **Emul8** and **Tracealyzer**

Johan Kraft, Percepio AB, johan.kraft@percepio.com

percepio
SENSING SOFTWARE

Virtual hardware?

- Use a system simulator as development platform
  - Instruction-level accurate and surprisingly fast
- Emul8
  - System simulator for embedded software development
  - Developed by Antmicro and Real-Time Embedded

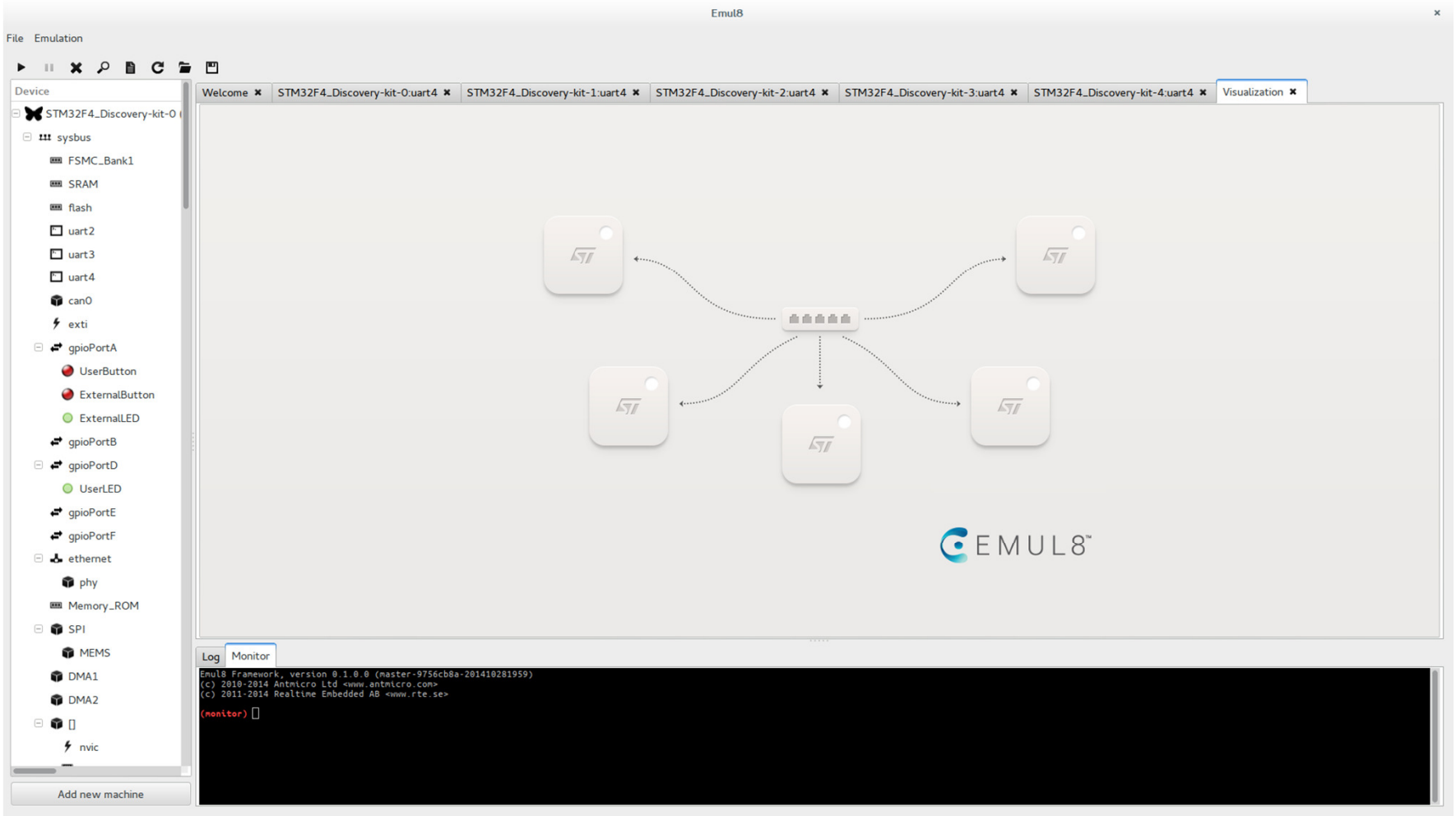
Why use virtual hardware platforms?

- Start software development earlier
- Developers not limited by shared hardware
- Simulate full system with multiple processors
- Full control
  - Everything is accessible and deterministic
  - Save system state and rewind/resume debugging
  - Deterministic automated testing on system level

Issues

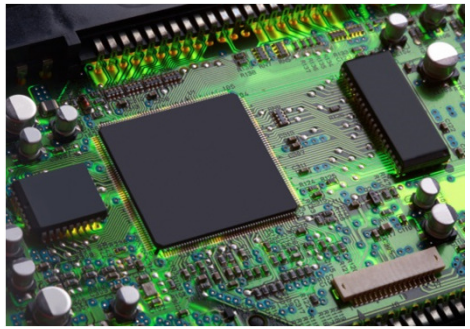
- Hardware models
  - Modular, parts can be reused
  - Model libraries available
- Accuracy
  - Approximate timing, not cycle-accurate
  - Deterministic – no ”random variations”
  - Insert timing variations to increase coverage?

Emul8


Emul8
File Emulation
Device
STM32F4_Discovery-kit-0
sysbus
FSMC_Bank1
SRAM
flash
uart2
uart3
uart4
can0
exti
gpioPortA
UserButton
ExternalButton
ExternalLED
gpioPortB
gpioPortD
UserLED
gpioPortE
gpioPortF
ethernet
phy
Memory_ROM
SPI
MEMS
DMA1
DMA2
[]
nvic
Add new machine
Welcome
STM32F4_Discovery-kit-0:uart4
STM32F4_Discovery-kit-1:uart4
STM32F4_Discovery-kit-2:uart4
STM32F4_Discovery-kit-3:uart4
STM32F4_Discovery-kit-4:uart4
Visualization
EMUL8
Log
Monitor
Emul8 Framework, version 0.1.0.0 (master-9756cb8a-201410281959)
(c) 2010-2014 Antmicro Ltd <www.antmicro.com>
(c) 2011-2014 Realtime Embedded AB <www.rte.se>
(monitor)

Tracing shows the run-time behavior

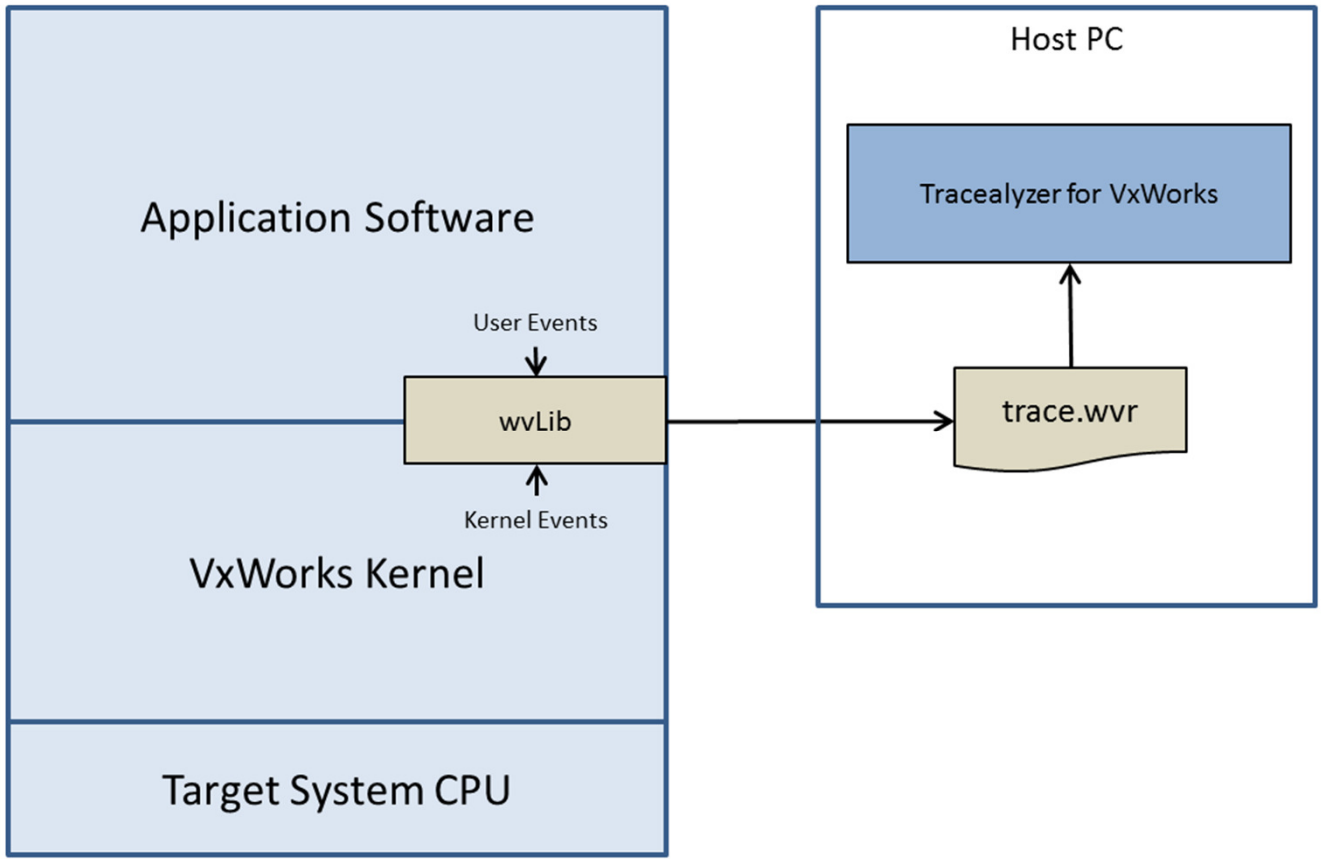
Record

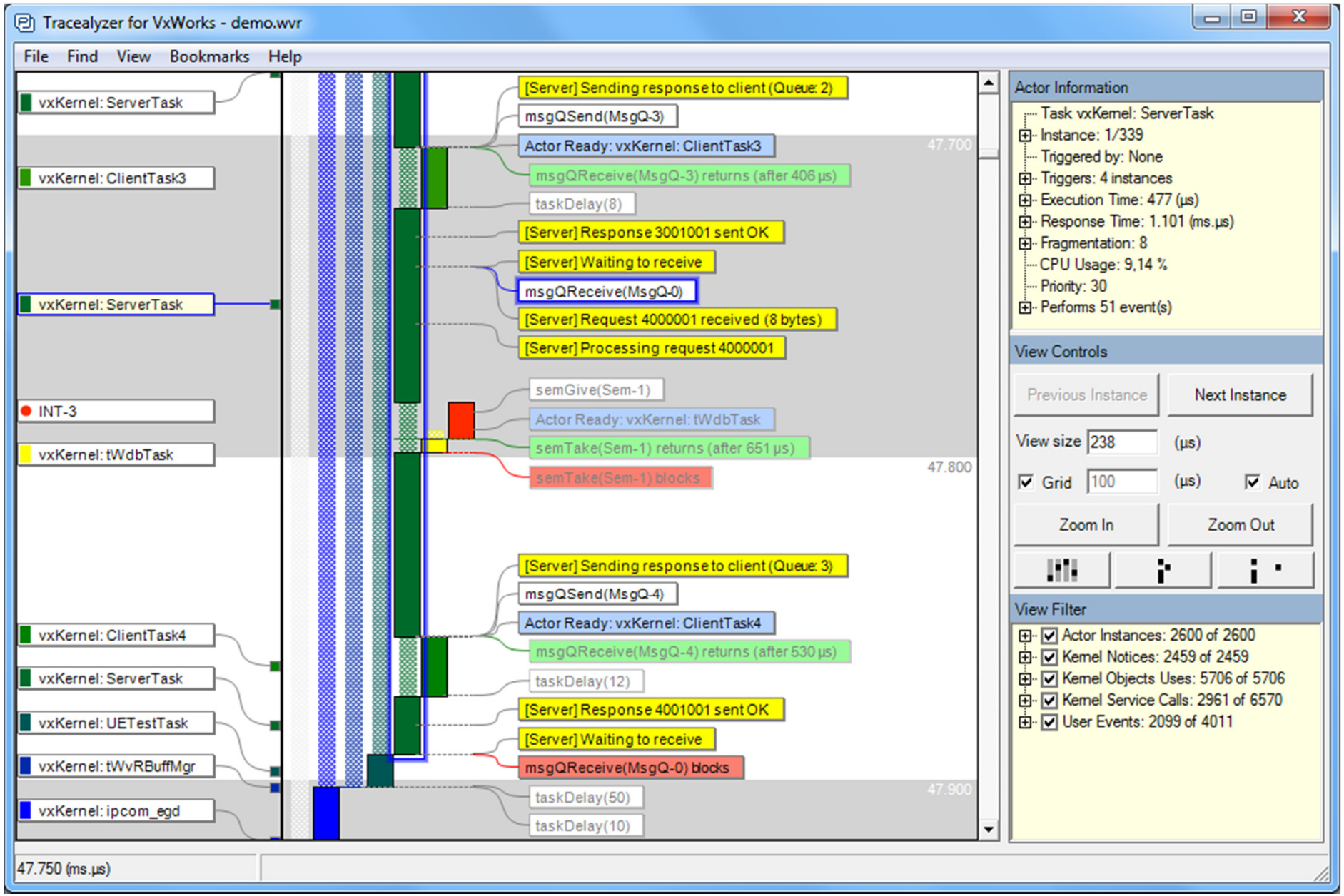


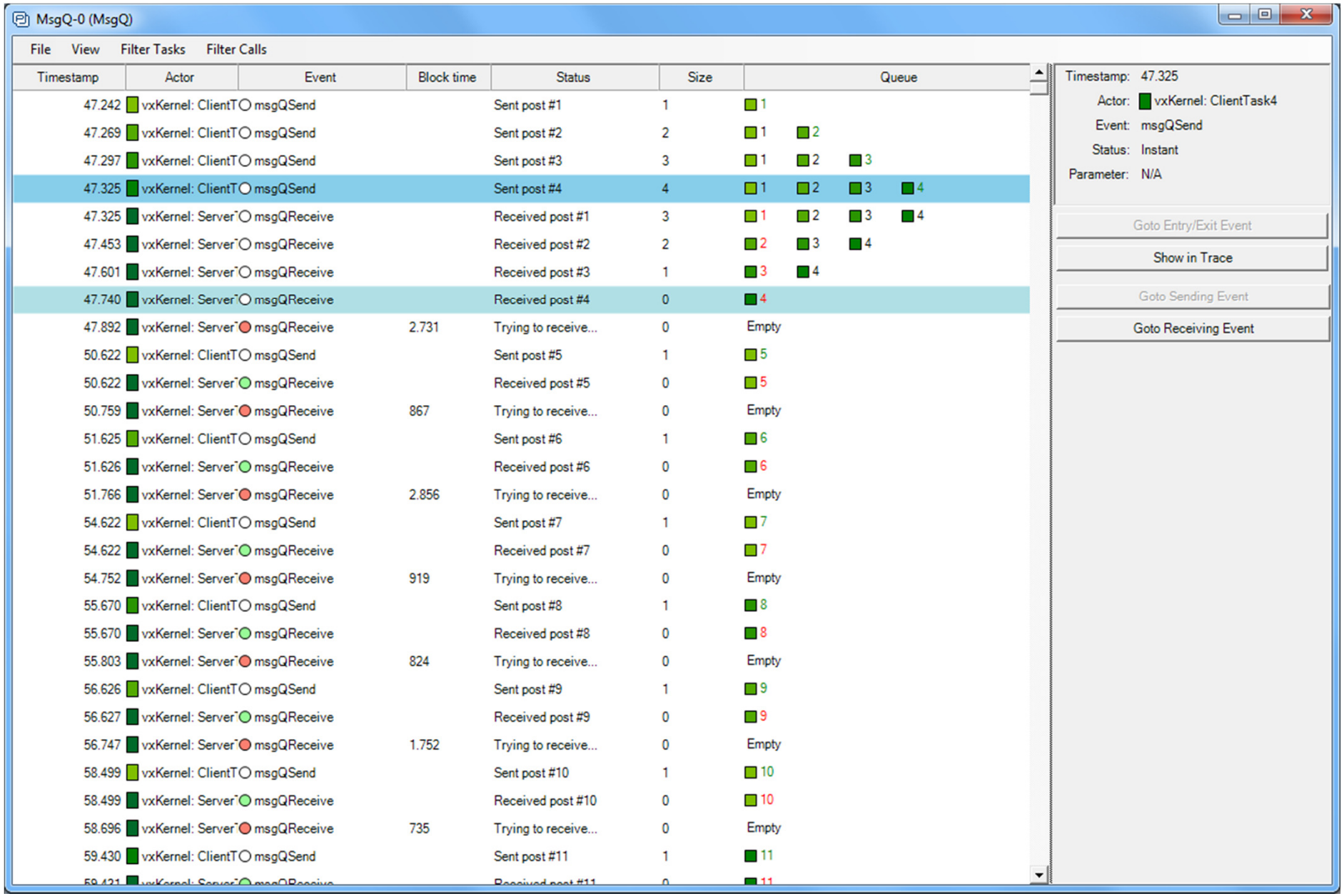
Software Behavior

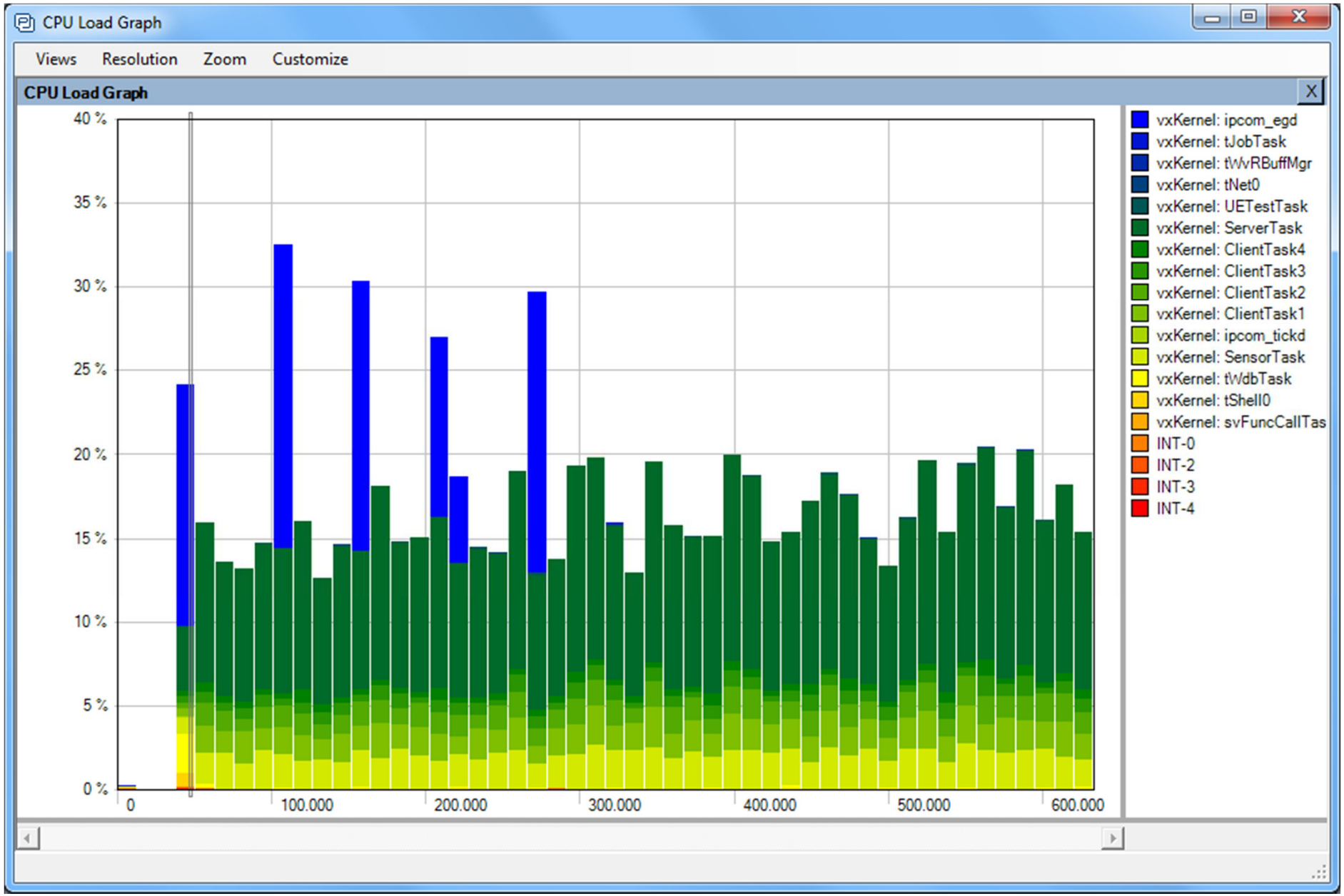
Visualize

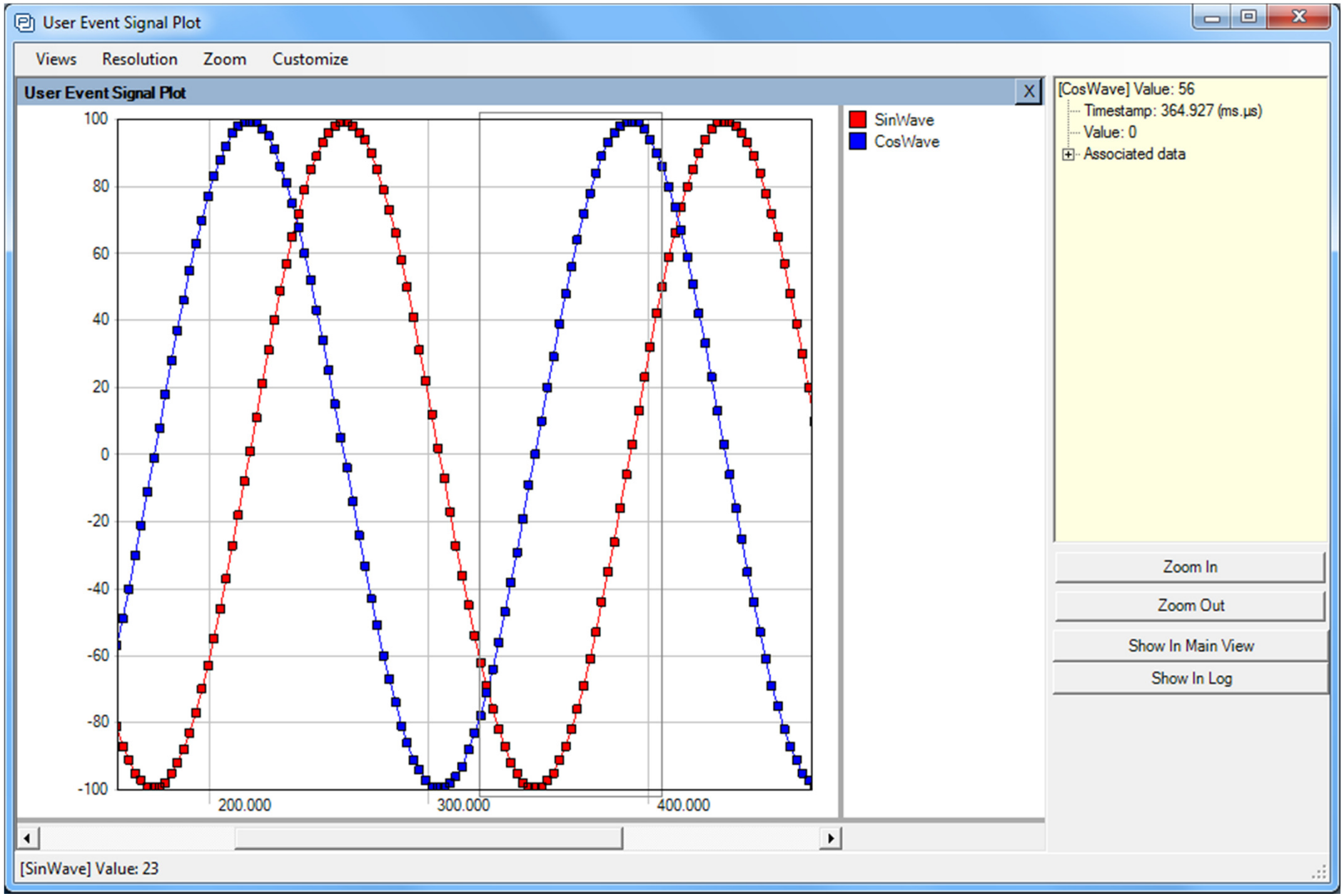
Software Trace

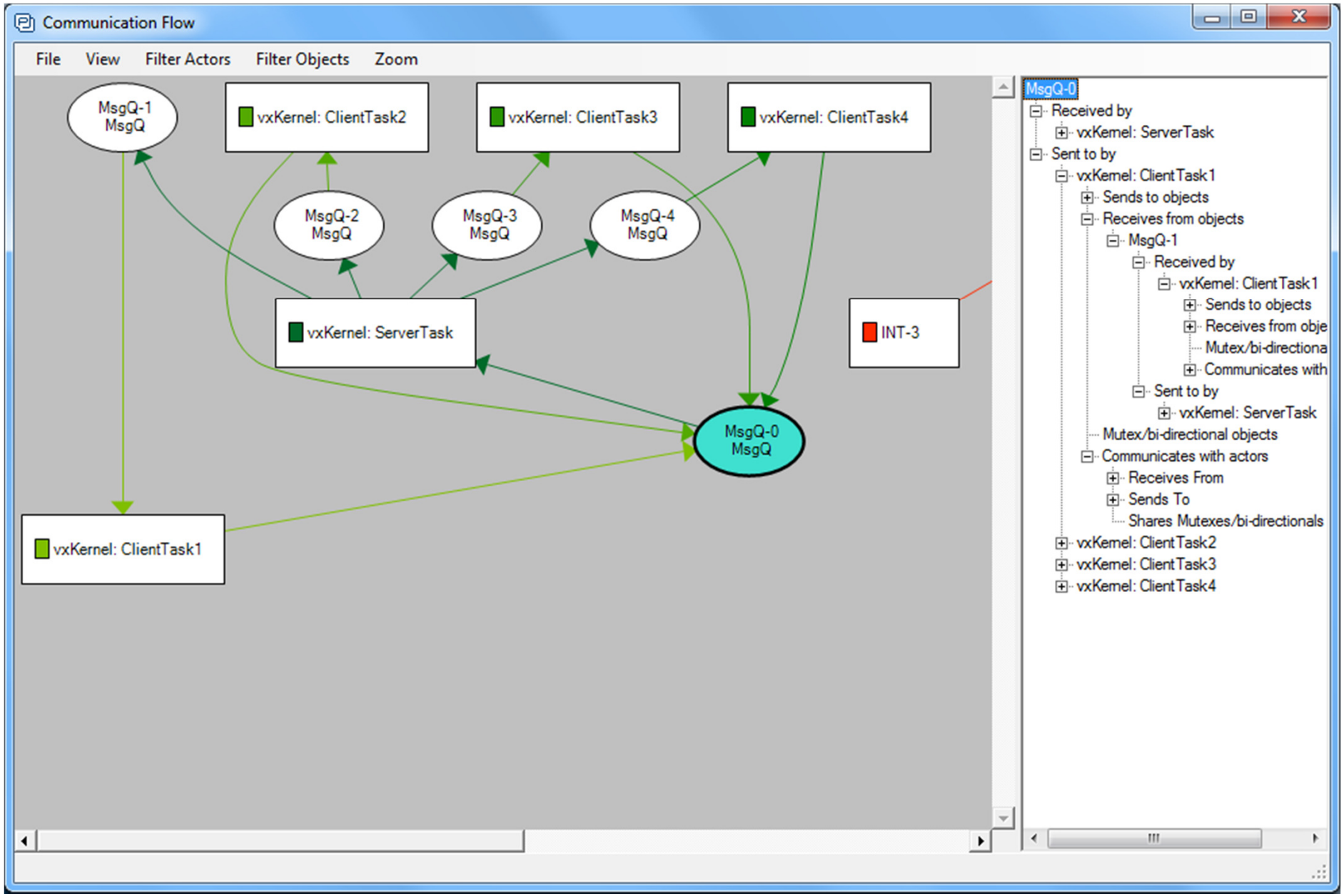



Tracealyzer for VxWorks - demo.wvr
File
Find
View
Bookmarks
Help
vxKernel: ServerTask
vxKernel: ClientTask3
vxKernel: ServerTask
INT-3
vxKernel: tWdbTask
vxKernel: ClientTask4
vxKernel: ServerTask
vxKernel: UETestTask
vxKernel: tWvRBuffMgr
vxKernel: ipcom_egd
[Server] Sending response to client (Queue: 2)
msgQSend(MsgQ-3)
Actor Ready: vxKernel: ClientTask3
47.700
msgQReceive(MsgQ-3) returns (after 406 µs)
taskDelay(8)
[Server] Response 3001001 sent OK
[Server] Waiting to receive
msgQReceive(MsgQ-0)
[Server] Request 4000001 received (8 bytes)
[Server] Processing request 4000001
semGive(Sem-1)
Actor Ready: vxKernel: tWdbTask
semTake(Sem-1) returns (after 651 µs)
47.800
semTake(Sem-1) blocks
[Server] Sending response to client (Queue: 3)
msgQSend(MsgQ-4)
Actor Ready: vxKernel: ClientTask4
msgQReceive(MsgQ-4) returns (after 530 µs)
taskDelay(12)
[Server] Response 4001001 sent OK
[Server] Waiting to receive
msgQReceive(MsgQ-0) blocks
47.900
taskDelay(50)
taskDelay(10)
47.750 (ms.µs)
Actor Information
Task vxKernel: ServerTask
Instance: 1/339
Triggered by: None
Triggers: 4 instances
Execution Time: 477 (µs)
Response Time: 1.101 (ms.µs)
Fragmentation: 8
CPU Usage: 9,14 %
Priority: 30
Performs 51 event(s)
View Controls
Previous Instance
Next Instance
View size 238 (µs)
Grid 100 (µs) Auto
Zoom In
Zoom Out
View Filter
Actor Instances: 2600 of 2600
Kernel Notices: 2459 of 2459
Kernel Objects Uses: 5706 of 5706
Kernel Service Calls: 2961 of 6570
User Events: 2099 of 4011

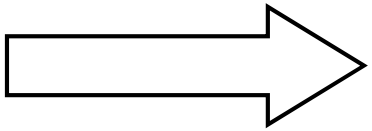

MsgQ-0 (MsgQ)
File
View
Filter Tasks
Filter Calls
Timestamp
Actor
Event
Block time
Status
Size
Queue
47.242 vxKernel: ClientT msgQSend Sent post #1 1 1
47.269 vxKernel: ClientT msgQSend Sent post #2 2 1 2
47.297 vxKernel: ClientT msgQSend Sent post #3 3 1 2 3
47.325 vxKernel: ClientT msgQSend Sent post #4 4 1 2 3 4
47.325 vxKernel: Server msgQReceive Received post #1 3 1 2 3 4
47.453 vxKernel: Server msgQReceive Received post #2 2 2 3 4
47.601 vxKernel: Server msgQReceive Received post #3 1 3 4
47.740 vxKernel: Server msgQReceive Received post #4 0 4
47.892 vxKernel: Server msgQReceive 2.731 Trying to receive... 0 Empty
50.622 vxKernel: ClientT msgQSend Sent post #5 1 5
50.622 vxKernel: Server msgQReceive Received post #5 0 5
50.759 vxKernel: Server msgQReceive 867 Trying to receive... 0 Empty
51.625 vxKernel: ClientT msgQSend Sent post #6 1 6
51.626 vxKernel: Server msgQReceive Received post #6 0 6
51.766 vxKernel: Server msgQReceive 2.856 Trying to receive... 0 Empty
54.622 vxKernel: ClientT msgQSend Sent post #7 1 7
54.622 vxKernel: Server msgQReceive Received post #7 0 7
54.752 vxKernel: Server msgQReceive 919 Trying to receive... 0 Empty
55.670 vxKernel: ClientT msgQSend Sent post #8 1 8
55.670 vxKernel: Server msgQReceive Received post #8 0 8
55.803 vxKernel: Server msgQReceive 824 Trying to receive... 0 Empty
56.626 vxKernel: ClientT msgQSend Sent post #9 1 9
56.627 vxKernel: Server msgQReceive Received post #9 0 9
56.747 vxKernel: Server msgQReceive 1.752 Trying to receive... 0 Empty
58.499 vxKernel: ClientT msgQSend Sent post #10 1 10
58.499 vxKernel: Server msgQReceive Received post #10 0 10
58.696 vxKernel: Server msgQReceive 735 Trying to receive... 0 Empty
59.430 vxKernel: ClientT msgQSend Sent post #11 1 11
Timestamp: 47.325
Actor: vxKernel: ClientTask4
Event: msgQSend
Status: Instant
Parameter: N/A
Goto Entry/Exit Event
Show in Trace
Goto Sending Event
Goto Receiving Event


CPU Load Graph
Views
Resolution
Zoom
Customize
CPU Load Graph
X
40 %
35 %
30 %
25 %
20 %
15 %
10 %
5 %
0 %
0
100.000
200.000
300.000
400.000
500.000
600.000
vxKernel: ipcom_egd
vxKernel: tJobTask
vxKernel: tWvRBuffMgr
vxKernel: tNet0
vxKernel: UETestTask
vxKernel: ServerTask
vxKernel: ClientTask4
vxKernel: ClientTask3
vxKernel: ClientTask2
vxKernel: ClientTask1
vxKernel: ipcom_tickd
vxKernel: SensorTask
vxKernel: tWdbTask
vxKernel: tShell0
vxKernel: svFuncCallTas
INT-0
INT-2
INT-3
INT-4


User Event Signal Plot
Views
Resolution
Zoom
Customize
User Event Signal Plot
X
100
80
60
40
20
0
-20
-40
-60
-80
-100
200.000
300.000
400.000
SinWave
CosWave
[CosWave] Value: 56
Timestamp: 364.927 (ms.µs)
Value: 0
Associated data
Zoom In
Zoom Out
Show In Main View
Show In Log
[SinWave] Value: 23


Communication Flow
File
View
Filter Actors
Filter Objects
Zoom
MsgQ-1 MsgQ
vxKernel: ClientTask2
vxKernel: ClientTask3
vxKernel: ClientTask4
MsgQ-2 MsgQ
MsgQ-3 MsgQ
MsgQ-4 MsgQ
vxKernel: ServerTask
INT-3
MsgQ-0 MsgQ
vxKernel: ClientTask1
MsgQ-0
Received by
vxKernel: ServerTask
Sent to by
vxKernel: ClientTask1
Sends to objects
Receives from objects
MsgQ-1
Received by
vxKernel: ClientTask1
Sends to objects
Receives from obje
Mutex/bi-directiona
Communicates with
Sent to by
vxKernel: ServerTask
Mutex/bi-directional objects
Communicates with actors
Receives From
Sends To
Shares Mutexes/bi-directionals
vxKernel: ClientTask2
vxKernel: ClientTask3
vxKernel: ClientTask4

Tracing on virtual hardware platforms



System simulator - extended with OS-aware tracing!

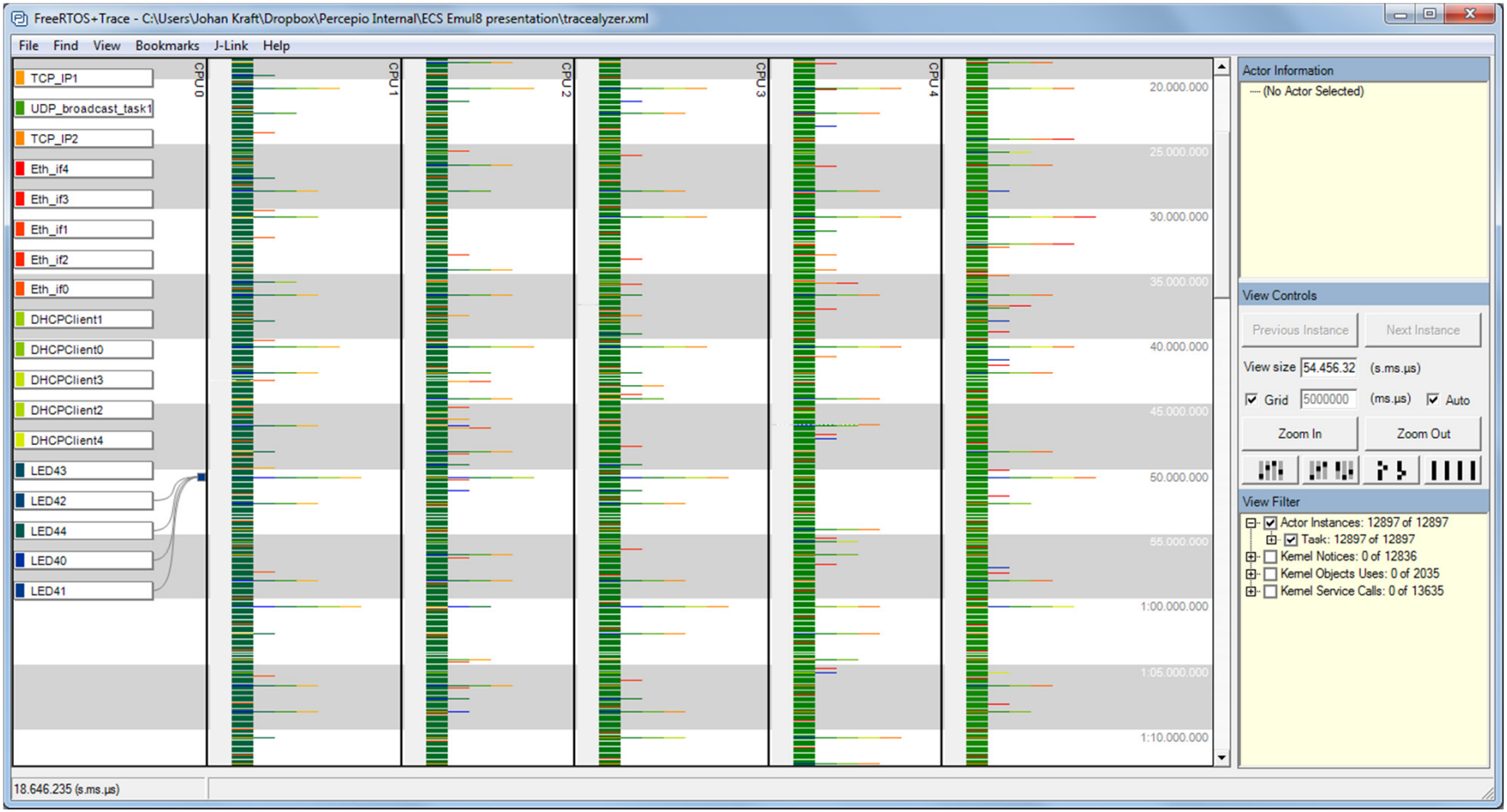
Comparison

	Hardware Trace	Software Trace	Simulator Trace
Max. level of detail	Instruction-level	Selected software events	Instruction-level
Impact/Overhead	None	CPU and RAM usage	None
What can be recorded	Hardware dependent (control and/or data)	Any software event	Any simulator event (software or hardware)
Hardware requirements	Trace-enabled CPU, board and debugger	None	None

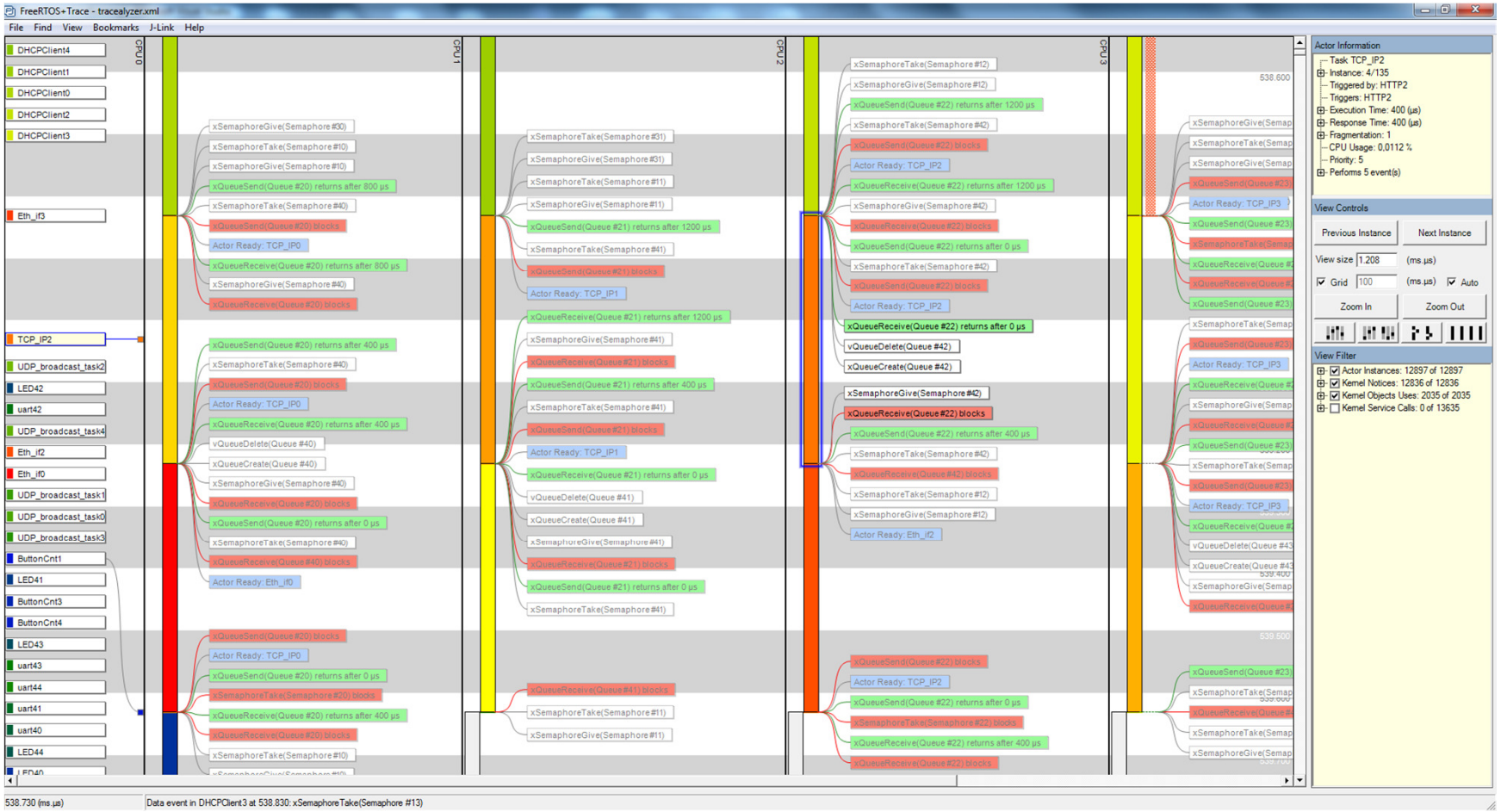
Prototype implementation

- Emul8 extended with
  - FreeRTOS awareness
  - Tracealyzer trace output (OS events)
- Demonstration system
  - 5 x STM32F4xx microcontrollers
  - FreeRTOS and TCP/IP communication
  - Physically implemented by RTE and Antmicro

Emul8 in Tracealyzer - Zoomed out



Emul8 in Tracealyzer - Zoomed in



Emul8 in Tracealyzer - Other views…

Thanks for listening!

Johan Kraft, Percepio AB,
johan.kraft@percepio.com
073-3196971